

# Sprint 1 (26.03 – 12.04)

In diesem Dokument wird das Vorgehen für den ersten Sprint des Projektes FDZ<sup>1</sup> Transport Simulation dargestellt. Es werden die Ziele für den Sprint aufgelistet, der Sprint Backlog vorgestellt und die darin enthaltenen Aufgaben, soweit möglich, nach den Vorgaben der Auftraggeber priorisiert.

## Ziele

Der erste Sprint des Projektes kann als Sonderfall angesehen werden. Hier muss nicht auf die Ereignisse in den letzten Sprints eingegangen werden, sondern stattdessen die Richtung für das gesamte Projekt vorgegeben werden. Ziel ist es also, Rücksprache über die Anforderungen mit den Auftraggebern zu halten, um sich anschließend in die vorhandene Dokumentation der FDZ einzuarbeiten, damit abschließend ein Lastenheft entworfen werden kann, das in seinem Umfang den Forderungen der Auftraggeber entspricht.

## Backlog

Das große Ziel dieses Sprints ist die Einarbeitung in das Projekt und die Sammlung der Anforderungen, um in Sprint 2 ein umfangreiches Backlog erstellen zu können, welches vom Auftraggeber priorisiert werden kann.

Nr.	Beschreibung	Priorität
1	Sammeln von Fragen für das Interview mit Auftraggebern	Hoch
2	Treffen mit Auftraggebern	Hoch
3	Sammeln der Anforderungen in einem vorläufigen Dokument	Mittel
4	Einarbeitung in die vorhandene Dokumentation der FDZ	Mittel
5	Erstellen von Wireframes als Anhaltspunkt für die grafische Oberfläche	Niedrig
6	Entwurf des Lastenhefts	Mittel

## Allgemeine Anmerkungen

Das Team hatte zu Anfang des Sprints die Entscheidung getroffen, das Vorgehensmodell Scrum zu wählen. Diese Entscheidung beruht vor Allem auch auf der Rücksprache mit den Auftraggebern, welche dem Team die Möglichkeit eingeräumt haben, sich wöchentlich mit einem Vertreter und Zuständigen der FDZ zu treffen, um mögliche Probleme zu besprechen und Anforderungen zu spezifizieren, bevor die jeweilige Implementierung der Problempunkte beginnt.

---

<sup>1</sup> FDZ – Fabrik der Zukunft

# Review Sprint 1

Im Folgenden wird der erste Sprint rückblickend analysiert und bewertet. Die Bewertungsgrundlage bilden hier die erledigten Aufgaben im Vergleich zu den vorhandenen, sowie die Zufriedenheit des Auftraggebers mit den fertigen Aufgaben aus dem Backlog.

## Aufgaben

In der Tabelle wird das komplette Backlog aus Sprint 1 gezeigt. Zusätzlich wird darin gekennzeichnet, welche Aufgaben erledigt wurden und mit welcher Zufriedenheit der Auftraggeber sie bewertet hat.

Nr.	Beschreibung	Priorität	Fertig am	Anmerkungen des Auftraggebers
1	Sammeln von Fragen für das Interview mit Auftraggebern	Hoch	26.03	-
2	Treffen mit Auftraggebern	Hoch	27.03	Befürwortung des Vorgehensmodells Scrum
3	Sammeln der Anforderungen in einem vorläufigen Dokument	Mittel	28.03	-
4	Einarbeitung in die vorhandene Dokumentation der FDZ	Mittel	30.03	-
5	Erstellen von Wireframes als Anhaltspunkt für die grafische Oberfläche	Niedrig	11.04	Visualisierung der Transportbänder anschaulich halten, keine UML-ähnlichen Ansichten auf die Fabrik
6	Entwurf des Lastenhefts	Mittel	11.04	Alle Anforderungen aufgenommen, Genaueres wird bei Bearbeitung besprochen
Erledigt				
Nicht erledigt				

Es wurden alle Aufgaben im Backlog erfüllt. Dieser Fall wird ein Ausnahmefall sein, denn im der Regel werden alle absehbaren Aufgaben im Backlog gesammelt, unabhängig davon, ob sie im nächsten Sprint machbar sind oder nicht.

## Rückblick

Der erste Sprint war der Zeitraum, in dem es darum ging, sich im Team einzuarbeiten, den Auftraggeber und das Projekt kennenzulernen und die Anforderungen zu sammeln, die das Team im Laufe des Projektes erfüllen muss. Im Allgemeinen ist dies gut gelungen, es ist ein hervorragendes Verständnis für die Funktionsweise der Fabrik bei den Entwicklern aufgetreten und die

Kommunikation mit den Auftraggebern, besonders mit dem wissenschaftlichen Mitarbeiter Alexander Schmid, ist sehr hilfreich.

Das Team ist zuversichtlich, das Projekt im vorgesehenen Zeitraum so zu beenden, dass alle Teilhaber zufrieden sind.

## Sprint 2 (12.04 – 19-04)

Auf den folgenden Seiten wird das Vorgehen für Sprint 2 erläutert. Dieser wird der erste Sprint sein, in dem fertiger Programm-Code geschrieben wird. Deshalb wurden vor dem abschließenden Gespräch mit Alexander Schmid mögliche Aufgaben gesammelt, die in Sprint 2 erledigt werden könnten. Diese Aufgaben wurden dann priorisiert.

### Ziele

Das Ziel für Sprint 2 ist es, die Hauptaufgaben des Programms zu analysieren und dementsprechende Analysen wie User-Stories und UML-Diagramme zu entwerfen.

### Backlog

Der Backlog in Sprint 2 wird einige Aufgaben enthalten, die im Allgemeinen erledigt werden müssen, allerdings nicht zwingend zu den Zielen des Sprints passen. Die Aufgaben wurden erstmals vom Auftraggeber priorisiert. In Sprint 1 erledigte Aufgaben werden hier nicht wieder aufgeführt.

Nr.	Beschreibung	Priorität
1	US Systemstart	Hoch
2	US Anforderung leerer Schlitten	Hoch
3	US Neupositionierung eines Schlittens	Hoch
4	US Freigabe eines Schlittens/Palette	Hoch
5	Basisarchitektur	Mittel
6	Sequenzdiagramm Abarbeitung empfangener Nachricht	Mittel
7	Implementierung der Klassen	Niedrig
8	Parse der Nachricht	Niedrig
9	Validierung der Werte	Niedrig
10	Implementierung RequestEmptyCarriage	Niedrig
11	Implementierung ReleaseCarriage	Niedrig
12	Implementierung RepositionCarriage	Niedrig
13	Implementierung ShutdownTransport	Niedrig

### Allgemeine Anmerkungen

Nach Abschluss von Sprint 2 soll die Analyse zu großen Teilen abgeschlossen sein. Die restlichen offenen Analysen sollen dann vor der Implementierung der einzelnen Aufgaben entstehen. Demnach soll es nach Sprint 2 möglich sein, die Grundstruktur des Programms ohne weiteres zu implementieren.

## Review Sprint 2

Im Folgenden wird der zweite Sprint rückblickend analysiert und bewertet. Die Bewertungsgrundlage bilden hier die erledigten Aufgaben im Vergleich zu den vorhandenen, sowie die Zufriedenheit des Auftraggebers mit den fertigen Aufgaben aus dem Backlog.

### Aufgaben

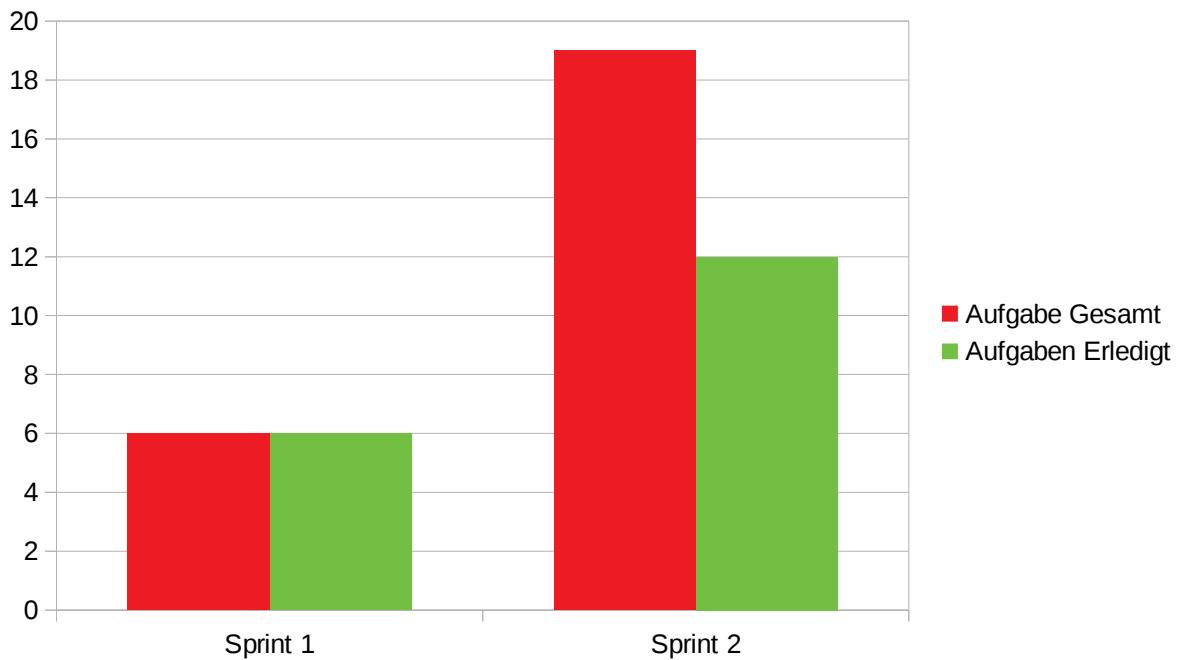
Die folgende Liste beschreibt die Aufgaben, die im Backlog vermerkt waren, wann sie erledigt wurden und welche Anmerkungen die Auftraggeber zum Ergebnis hatten.

Nr.	Beschreibung	Priorität	Fertig am	Anmerkungen des Auftraggebers
1	US Systemstart	Hoch	-	
2	US Anforderung leerer Schlitten	Hoch	-	
3	US Neupositionierung eines Schlittens	Hoch	-	
4	US Freigabe eines Schlittens/Palette	Hoch	-	
5	Basisarchitektur	Mittel	-	
6	Sequenzdiagramm Abarbeitung empfangener Nachricht	Mittel		Die Idee der Abarbeitung ist gut, jedoch wird der korrekte Ablauf der Kommandos nicht garantiert, wenn ein Kommando pro Thread ausgeführt wird. Deshalb Speicherung der Kommandos in einer Queue.
Erledigt				
Nicht erledigt				
7	Implementierung der Klassen	Niedrig	-	-
8	Parse der Nachricht	Niedrig	-	-
9	Validierung der Werte	Niedrig	-	-
10	Implementierung RequestEmptyCarriage	Niedrig	-	-
11	Implementierung ReleaseCarriage	Niedrig	-	-
12	Implementierung RepositionCarriage	Niedrig	-	-
13	Implementierung ShutdownTransport	Niedrig	-	-

Wie bereits im ersten Sprint angemerkt konnten nicht alle Aufgaben erfüllt werden. Jedoch ist es dem Team gelungen alle hoch priorisierten Aufgaben zu erledigen.

## Rückblick

Der zweite Sprint war der bis jetzt produktivste, es wurden weiterführende Analysen betrieben, die es ermöglichen, einen Großteil der Anforderungen ab dem 3. Sprint zu implementieren. Eine Übersicht der Aufgaben ist im folgenden Diagramm zu sehen.



Zu diesem Zeitpunkt ist das Diagramm zwar noch nicht sehr aussagekräftig, jedoch ist bereits zu erkennen, dass die Aufgaben in befriedigendem Maße erledigt wurden. Lediglich die Implementierungen fehlen. Anzumerken ist allerdings, dass die Anzahl der Aufgaben im Laufe des Projektes ständig steigen werden.

Rückblickend ist das Team immer noch sehr gut im Zeitplan, die Kommunikation mit den Auftraggebern ist nach wie vor sehr gut, bei Fragen ist immer ein Ansprechpartner, in der Regel Alexander Schmid, zur Verfügung.

## Sprint 3 (19.04 – 03.05)

Der 3. Sprint soll die Phase der Implementierung einleiten. Diese Phase soll große Teile der Logik abdecken, während das grafische Interface noch nicht weiter betrachtet wird.

### Ziele

Das Hauptziel in Sprint 3 ist es, die Kernlogik des Parsers der Simulation zu implementieren. Dieser bekommt empfangene Kommandos und zerlegt diese in Einzelteile, welche dann validiert und ausgeführt werden können. In Sprint 3 soll dieser Teil implementiert werden, um Konflikte früh mit den Auftraggebern zu besprechen, damit andere Herangehensweisen früh im Projekt analysiert werden können.

### Backlog

Nr.	Beschreibung	Priorität
1	Implementierung der Klassen	Hoch
2	Parsen der Nachricht	Hoch
3	Validierung der Werte	Hoch
4	Implementierung RequestEmptyCarriage	Hoch
5	Implementierung ReleaseCarriage	Hoch
6	Implementierung RepositionCarriage	Hoch
7	Implementierung ShutdownTransport	Hoch
8	Pflichtenheft	Mittel
9	Finales Screendesign	Mittel
10	Abfragedialog Wiederherstellung	Niedrig
11	Logmeldungen	Niedrig
12	Zustand laden, speichern	Mittel
13	Suchen einer Strecke bei Befehlsausführung	Hoch
14	Prüfung ob Palette mit ID vorhanden	Hoch
15	Prüfung ist leerer Schlitten in Station	Hoch
16	Station einfügen, löschen	Niedrig
17	Testen des Parsers	Niedrig
18	Analyse der Netzwerkverbindung	Niedrig
19	View erstellen	Niedrig
20	Controller erstellen	Niedrig
21	Speichern der optischen Elemente	Niedrig
22	View einstellen der Verbindungen	Niedrig

## Allgemeine Anmerkungen

Nach Abschluss des 3. Sprint soll es möglich sein, das Parsen der Netzwerknachrichten, zu diesem Zeitpunkt simuliert durch Testeingaben, zu testen. Was alles abgedeckt sein sollte, ist das Suchen der Stationen anhand der Positionsangaben in der Netzwerknachricht, das finden von Schlitten IDs und das Prüfen, ob eine Verbindung zwischen zwei Stationen ohne Unterbrechung möglich ist.

# Review Sprint 3

Der Sprint hatte die erste große Aufgabe des Programms inne, das Parsen der Netzwerknachricht. Dies ist dem Team auch gelungen, erste Tests haben bestätigt, dass die Nachrichten fehlerfrei geparsed werden und ungültige Nachrichten auch erkannt werden.

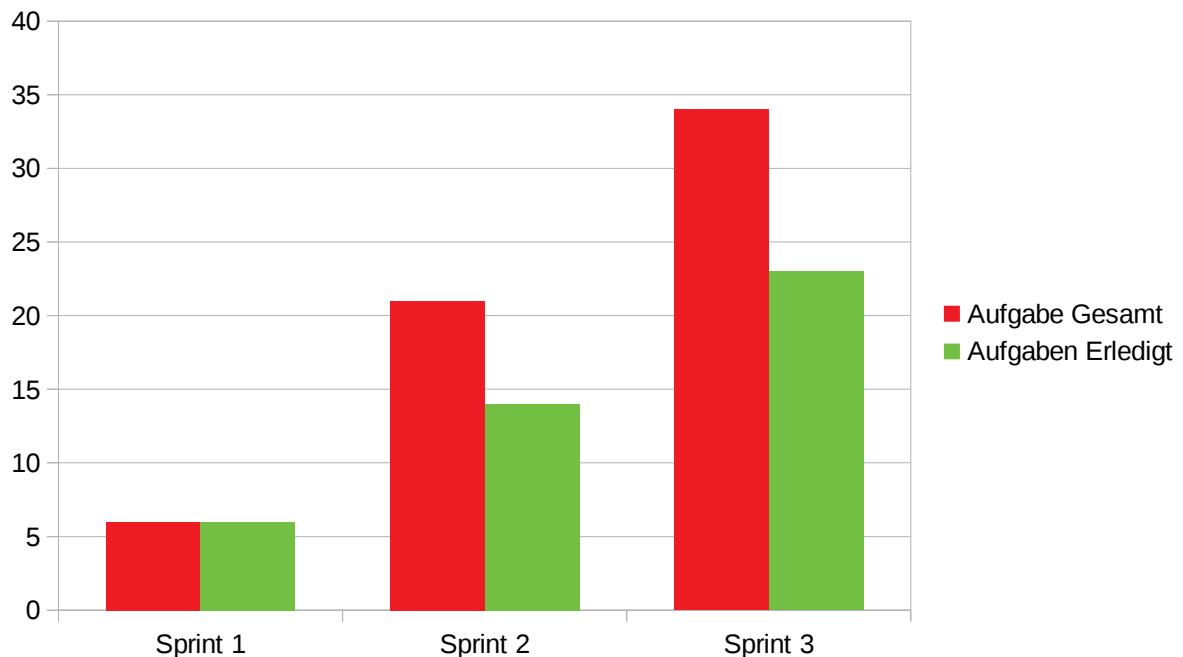
## Aufgaben

Die folgende Liste beschreibt die Aufgaben, die im Backlog vermerkt waren, wann sie erledigt wurden und welche Anmerkungen die Auftraggeber zum Ergebnis hatten.

Nr.	Beschreibung	Priorität	Fertig am	Anmerkungen des Auftraggebers
1	Implementierung der Klassen	Hoch		
2	Parsen der Nachricht	Hoch		
3	Validierung der Werte	Hoch		
4	Implementierung RequestEmptyCarriage	Hoch		
5	Implementierung ReleaseCarriage	Hoch		
6	Implementierung RepositionCarriage	Hoch		
7	Implementierung ShutdownTransport	Hoch		
8	Suchen einer Strecke bei Befehlsausführung	Hoch		
9	Prüfung ob Palette mit ID vorhanden	Hoch		
10	Prüfung ist leerer Schlitten in Station	Hoch		
8	Pflichtenheft	Mittel		
9	Finales Screendesign	Mittel		
Erledigt				
Nicht erledigt				
10	Station einfügen, löschen	Niedrig		
11	Testen des Parsers	Niedrig		
12	Analyse der Netzwerkverbindung	Niedrig		
13	View erstellen	Niedrig		
14	Controller erstellen	Niedrig		
15	Speichern der optischen	Niedrig		

	Elemente			
16	View einstellen der Verbindungen	Niedrig		
17	Abfragedialog Wiederherstellung	Niedrig		
18	Logmeldungen	Niedrig		
19	Zustand laden, speichern	Mittel		

## Rückblick



Langsam ist dem Diagramm zu entnehmen, dass in jedem Sprint eine gute Anzahl an Aufgaben erledigt werden. Auch ist nun zu sehen, dass im Verlauf des Projektes neue Aufgaben hinzukommen. Dennoch ist das Team immer noch gut im Zeitplan.

Die Stimmung im Team ist nach wie vor zuversichtlich, die Absprachen mit den Auftraggebern werden immer genauer und produktiver.

## Sprint 4 (04.05 – 24.05)

Im vierten Sprint werden die zusätzlichen Aufgaben implementiert, die neben der Grundfunktionalität der Simulation von den Auftraggebern gefordert wurden. Außerdem wird eine grafische Oberfläche für das Programm erstellt.

### Ziele

Das Hauptziel des vierten Sprints ist das Erstellen einer grafischen Oberfläche, damit im nächsten Sprint das Model mit der View verknüpft werden kann. Nebenbei werden einige Zusatz-Aufgaben der Funktionalität der Simulation implementiert.

### Backlog

Nr.	Beschreibung	Priorität
1	Controller erstellen	Hoch
2	View mit FX Scenebuilder erstellen	Hoch
3	View Einstellen der Verbindungen	Hoch
4	View Konfigurieren der Stationen	Hoch
5	View Kreuzungen einfügen	Hoch
6	Speichern der optischen Elemente	Hoch
7	Laden der optischen Elemente	Hoch
8	Zustand Laden/ Speichern	Niedrig
9	Abfragedialog Wiederherstellung	Niedrig
10	Logmeldungen	Niedrig
11	Station einfügen, löschen	Mittel
12	Testen des Parsers	Mittel
13	Analyse der Netzwerkverbindung	Mittel
14	Implementierung der Netzwerkverbindung	Mittel

### Allgemeine Anmerkungen

In Sprint 5 soll das Model und seine Kernlogik mit der View verbunden werden, damit ein fertiges Programm getestet werden kann.

# Review Sprint 4

In Sprint 4 lag der Fokus auf der View, außerdem auf Aufgaben, die neben der Grundfunktionalität zusätzlich von den Auftraggebern gefordert wurden.

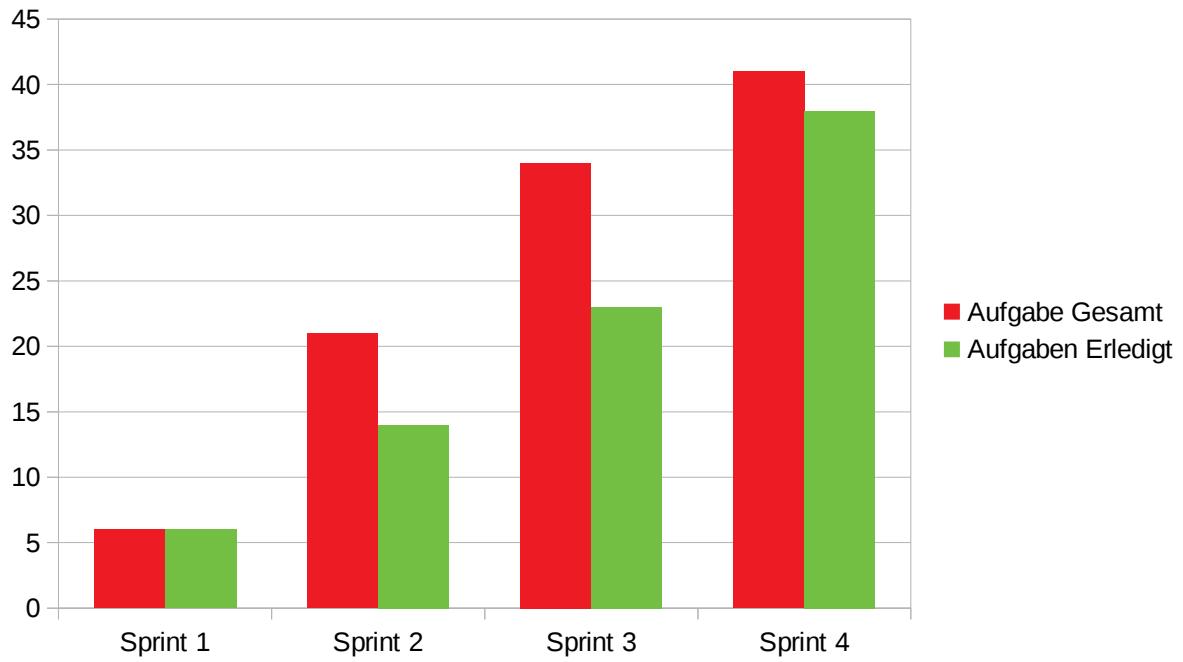
## Aufgaben

Die folgende Liste beschreibt die Aufgaben, die im Backlog vermerkt waren, wann sie erledigt wurden und welche Anmerkungen die Auftraggeber zum Ergebnis hatten.

Nr.	Beschreibung	Priorität	Fertig am	Anmerkungen des Auftraggebers
1	Controller erstellen	Hoch		
2	View mit FX Scenebuilder erstellen	Hoch		
3	View Einstellen der Verbindungen	Hoch		
4	View Konfigurieren der Stationen	Hoch		
5	View Kreuzungen einfügen	Hoch		
6	Speichern der optischen Elemente	Hoch		
7	Laden der optischen Elemente	Hoch		
8	Station einfügen, löschen	Mittel		
9	Testen des Parsers	Mittel		
10	Analyse der Netzwerkverbindung	Mittel		
11	Implementierung der Netzwerkverbindung	Mittel		
Erledigt				
Nicht erledigt				
12	Zustand Laden/ Speichern	Niedrig		
13	Abfragedialog Wiederherstellung	Niedrig		
14	Logmeldungen	Niedrig		

## Rückblick

Das Projekt neigt sich langsam dem Ende zu, es ist zu erkennen, dass die meisten Aufgaben vom Team implementiert wurden und von den Auftraggebern begutachtet wurden.



Das Diagramm spiegelt wieder, dass die meisten Aufgaben der Simulation erledigt, implementiert und nach bearbeitet wurden. Sprint 5 kann ich nun mit dem Verbinden der Einzelemente befassen, wonach abschließend nur noch getestet werden muss.

## Sprint 5 (25.05 – 16.06)

In den folgenden Abschnitten wird das Vorgehen für Sprint 5 erläutert. Wie im letzten Sprint angemerkt, sollen Model und View verbunden werden.

### Ziele

In Sprint 5 werden Model und View verbunden. Dazu benötigt das Model noch eine Fassade, die View noch einen Controller. Zusätzlich werden die letzten beiläufigen Funktionen des Programms implementiert, sodass ab Sprint 6 getestet werden kann.

### Backlog

Nr.	Beschreibung	Priorität
1	Zustand Laden/ Speichern	Niedrig
2	Abfragedialog Wiederherstellung	Niedrig
3	Logger Abändern und Einfügen	Mittel
4	Schreiben einer Fassade für Model	Hoch
5	Fertigstellen des Controllers mit Funktionen der Fassade	Hoch
6	Verbinden von Model und View	Hoch
7	Bugfixing bei Verbindung	Hoch
8	Observer für Änderung an Stationen	Mittel
9	Observer für Verbindungsstatus der Netzwerkschnittstelle	Mittel

### Allgemeine Anmerkungen

Das Programm sollte am Ende des Sprints vollständig funktionstüchtig sein, der Programm-Code soll abgegeben werden. In Sprint 6 soll anschließend nur getestet und gerefactored werden.

# Review Sprint 5

Im folgenden werden die Fortschritte in Sprint 5 dargestellt. Dieser stellt einen wichtigen Meilenstein im Projekt dar, da er die zwei Teile des Programms, Logik und optische Darstellung, vereint.

## Aufgaben

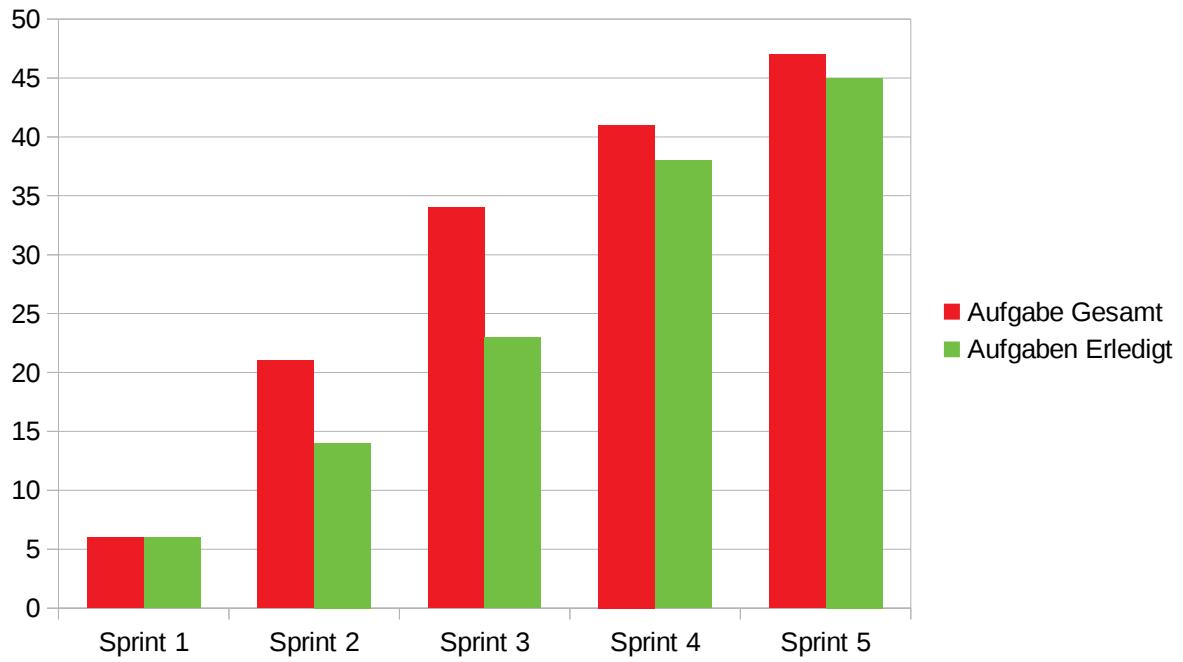
Die folgende Liste beschreibt die Aufgaben, die im Backlog vermerkt waren, wann sie erledigt wurden und welche Anmerkungen die Auftraggeber zum Ergebnis hatten.

Nr.	Beschreibung	Priorität	Fertig am	Anmerkungen des Auftraggebers
1	Logger Abändern und Einfügen	Mittel		
2	Schreiben einer Fassade für Model	Hoch		
3	Fertigstellen des Controllers mit Funktionen der Fassade	Hoch		
4	Verbinden von Model und View	Hoch		
5	Bugfixing bei Verbindung	Hoch		
6	Observer für Änderung an Stationen	Mittel		
7	Observer für Verbindungsstatus der Netzwerkschnittstelle	Mittel		
Erledigt				
Nicht erledigt				
8	Zustand Laden/ Speichern	Niedrig		
9	Abfragedialog Wiederherstellung	Niedrig		

## Rückblick

Die Hauptaufgaben aus Sprint 5 sind alle erledigt worden, jedoch ist es noch nicht gelungen, alle Funktionen zu implementieren. Dies muss in Sprint 6 nachgeholt werden. Zudem wurde nur oberflächlich getestet, ob alle Funktionen einwandfrei ablaufen, deshalb muss auch dies wie geplant in Sprint 6 durchgeführt werden.

Die allgemeine Stimmung im Projekt bleibt positiv, auch die Zustimmung über die Ergebnisse seitens der Auftraggeber bleibt erhalten.



Das Diagramm zeigt, dass die meisten Aufgaben erledigt wurden. Jedoch bedeutet das nicht, dass das Projekt schon zu Ende geht. In Sprint 6 wird erörtert, welche Tests dringend durchgeführt werden müssen.

## **Sprint 6 (17.06 – 05.07.‘18)**

Der 6. Sprint ist der abschließende Sprint im Projekt. Alle Anforderungen werden hier zu Ende gebracht.

### **Ziele**

In Sprint 6 ist es das Ziel, die Anforderungen der Auftraggeber lückenlos zu erfüllen. Abschließend wird nochmals ein Meeting mit Alexander Schmid gehalten, in dem die finalen Ergebnisse vorgestellt und nochmals geprüft werden. Zudem liegt der Schwerpunkt in Sprint 6 beim Testen und Debuggen.

### **Backlog**

Nr.	Beschreibung	Priorität
1	Zustand Laden/ Speichern	Hoch
2	Abfragedialog Wiederherstellung	Hoch
3	Tests Noah	Hoch
4	Tests Dennis	Hoch
5	Tests Andreas	Hoch

# Review Sprint 6

## Aufgaben

Die folgende Liste beschreibt die Aufgaben, die im Backlog vermerkt waren, wann sie erledigt wurden und welche Anmerkungen die Auftraggeber zum Ergebnis hatten.

Nr.	Beschreibung	Priorität	Fertig am	Anmerkungen des Auftraggebers
1	Zustand Laden/ Speichern	Hoch	23.06.'18	-
2	Abfragedialog Wiederherstellung	Hoch	21.03.'18	-
3	Tests Noah	Hoch	26.06.'18	-
4	Tests Dennis	Hoch	26.06.'18	-
5	Tests Andreas	Hoch	26.06.'18	-

**Erledigt**

**Nicht erledigt**

## Rückblick

Rückblickend ist in Sprint 6 anzumerken, dass alle Aufgaben erledigt wurden. Das Programm ist komplett den Anforderungen entsprechend implementiert worden und beim abschließenden Gespräch mit Auftraggeber Alexander Schmid sind keine weiteren Probleme aufgetreten.

